NUWC-NPT Technical Report 12,132
25 March 2013

Software Package for Optical Propagation in the Ocean

Lee E. Estes
Autonomous and Defensive Systems Department

WARFARE CENTERS
NEWPORT

Naval Undersea Warfare Center Division
Newport, Rhode Island

Approved for public release; distribution is unlimited.

PREFACE

This report was prepared under NWA 100000787263/0010, principal investigator Lee E. Estes (Code 8211).

The technical reviewer for this report was Lynn T. Antonelli (Code 1512).

Reviewed and Approved: 25 March 2013

James S. Griffin
Head, Autonomous and Defensive Systems Department

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OPM control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (*DD-MM-YYYY*)	2. REPORT TYPE	3. DATES COVERED (*From – To*)
25-03-2013		

4. TITLE AND SUBTITLE Software Package for Optical Propagation in the Ocean	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Lee E. Estes	5.d PROJECT NUMBER NWA 100000787263/0010 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Undersea Warfare Center Division 1176 Howell Street Newport, RI 02841-1708	8. PERFORMING ORGANIZATION REPORT NUMBER TR 12,132
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITOR'S ACRONYM 11. SPONSORING/MONITORING REPORT NUMBER

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report describes a Monte Carlo simulation of image point spread functions in an absorbing and scattering water environment. A parallel processing MATLAB software package that implements the simulation is presented.

15. SUBJECT TERMS

Optics   Light Propagation   Light Scattering   Optical Systems

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
(U)	(U)	(U)	SAR	21	Lee E. Estes
					19b. TELEPHONE NUMBER (*Include area code*) 401-832-4007

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

TABLE OF CONTENTS

Section **Page**

LIST OF ILLUSTRATIONS

Figure **Page**

SOFTWARE PACKAGE FOR
OPTICAL PROPAGATION IN THE OCEAN

1. INTRODUCTION

When light travels in water, it experiences beam steering, diffraction, scattering due to inhomogeneities along with polarization effects, and absorption. Full wave propagation and ray tracing software are available to determine the effects of relatively smooth variations in the water index of refraction caused by thermal, salinity, and pressure variations. This report describes a ray-based Monte Carlo software package that ignores polarization effects and simulates light propagation in the presence of absorption and scattering due to particles and molecular fluctuations. While the main focus of the software is to produce a point spread function (PSF) that can be used to model imaging, the results include propagation times so that the bandwidth of the optical channel when used for communication is also generated. While some commercial software packages exist that accomplish these results, this report presents a slightly different point of view, as well as a parallel processing MATLAB software package that implements the Monte Carlo simulation.

2. THEORY

The model presented makes use of the following inherent optical properties to describe the optical environment:

$a \equiv$ optical absorption coefficient,

$\beta \equiv$ volume scattering coefficient,

$$b \equiv \text{optical scattering coefficient} = \int_{4\pi\, steradians} d\Omega\, \beta = \int_0^{2\pi} d\phi \int_0^{\pi} d\theta\, \sin(\theta)\, \beta(\phi,\theta). \quad (1)$$

Based on these definitions, a derived inherent optical property is the coherent beam attenuation coefficient, defined as

$$c = a + b\,. \quad (2)$$

All of the parameters defined in equations (1) and (2) are wavelength dependent.

In the case of a collimated laser beam with wavelength λ and power P_0, the coherent portion of the beam after propagating a distance R is given by

$$P(z) = P_0 \exp(-c_\lambda R). \tag{3}$$

In the case of a single wavelength point source with power P_0, the irradiance at a range R is given by

$$I(R) = P_0 \frac{\exp(-K_\lambda R)}{4\pi R^2}, \tag{4}$$

where K_λ is the diffuse attenuation coefficient, which is greater than the absorption coefficient because some of the light undergoes scattering before it arrives at a range R and must therefore travel a longer distance in the absorbing environment to arrive at R. The value of K_λ is determined by fitting the results generated by the Monte Carlo simulation of a point source to equation (4). Defined in this way, the value of K_λ increases slightly with range. The difference in travel range causes a corresponding spread in travel times. In addition, the point source light that arrives at R is spread over a range of angles because of the different scattering experience of each ray. This spread in angles is recorded in the Monte Carlo simulation and used to determine the image PSF seen by a camera located at range R and aimed at the point source.

The process outlined above is only an approximation of what occurs when a picture is taken of an object in the absorbing and scattering medium that lies between the camera and the object. For instance, it does not account for light that leaves the object and is backscattered to re-illuminate the object. It also does not account for light that leaves the object and reaches behind the camera to be scattered in front of the camera and then scattered into the camera. While these events occur, they are higher order events that are reasonably neglected.

Within the approximations described above, the process used to determine a PSF is presented in the flowchart in figure 1. As indicated in the third block of the flowchart, the initial ray direction is arbitrary for the assumed point source that demands statistical angular symmetry.

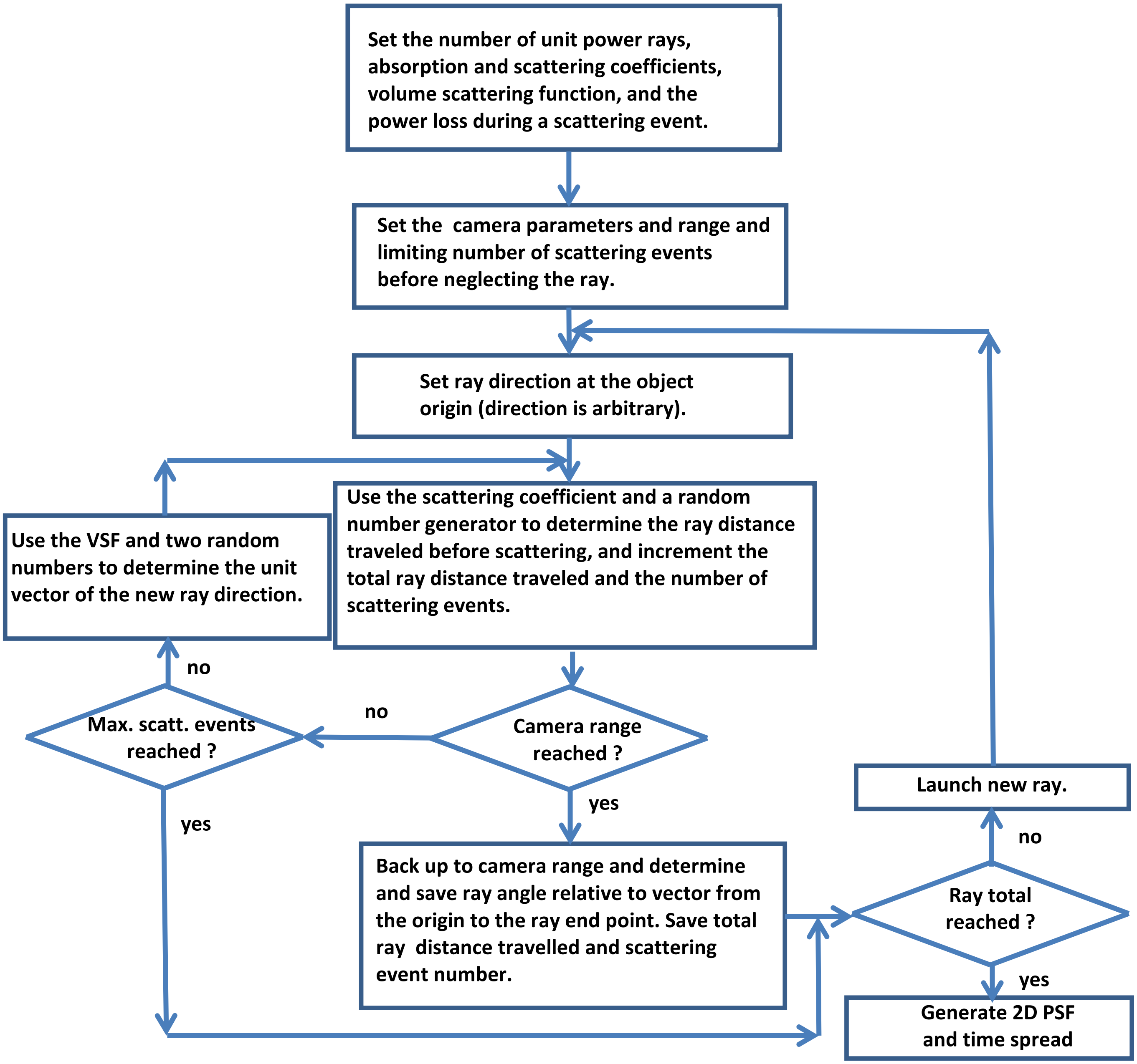

Figure 1. Flowchart for Simulation of Ray Propagation in the Presence of Scattering and Absorption

3. MATLAB CODE

The flowchart in figure 1 is implemented with three m-files that call two functions and import one mat-file that contains the needed volume scattering function data. The code uses parallel processing to determine the point spread function out to a selectable maximum range and at selected periodic intermediate ranges. Figure 2 depicts the software modules with their names, inputs, and outputs. Reference is made to the m-files presented in the appendix, where the input variables are defined at the beginning of each m-file. The contents of the output mat-files are defined in the two function and the PSF789parallel m-files.

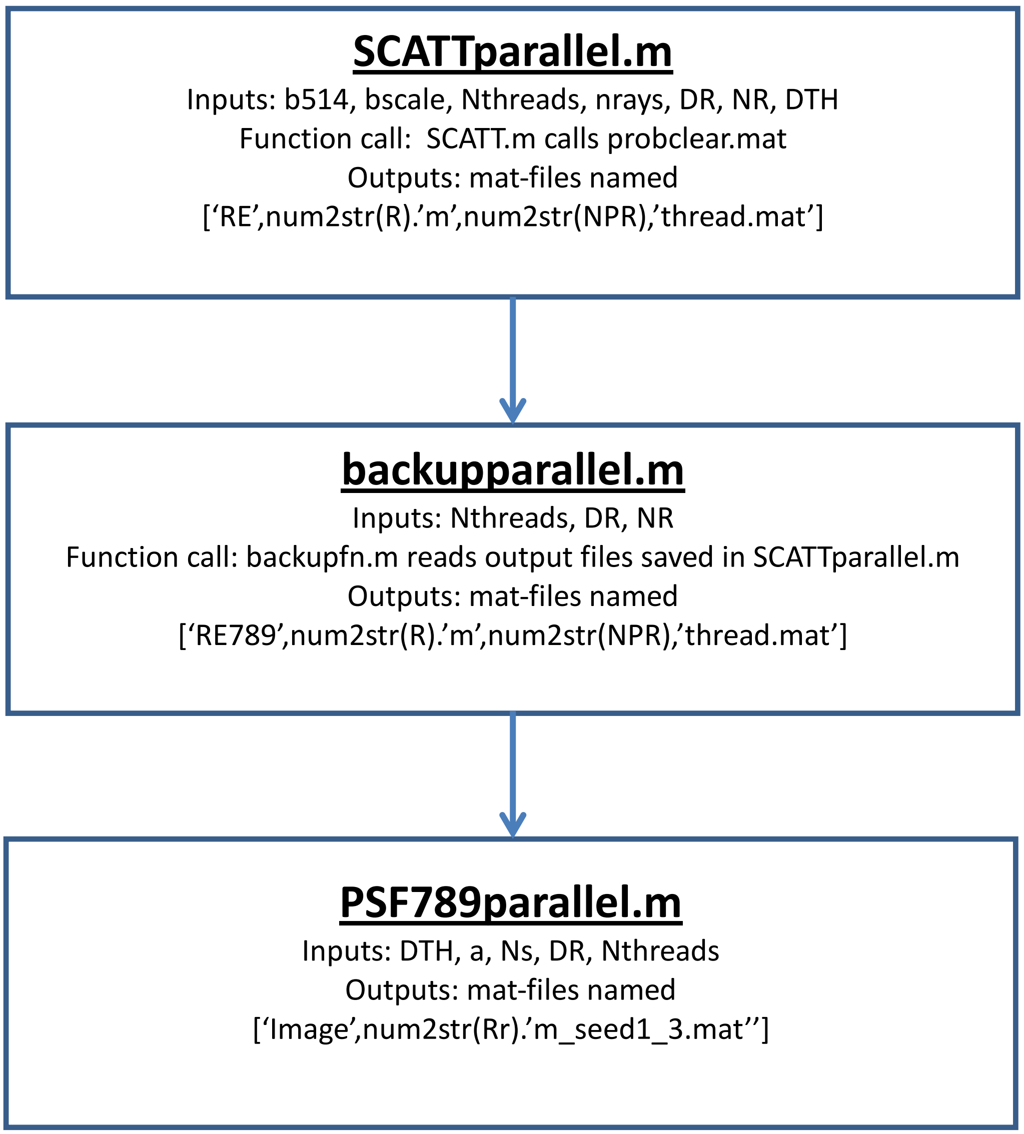

Figure 2. Parallel Processing MATLAB m-files Used to Implement the Monte Carlo Simulation Flow Charted in Figure 1 for a Periodic Sequence of Ranges

4. SUMMARY

The ray-based Monte Carlo software package described in this report ignores polarization effects and simulates light propagation in the presence of absorption and scattering due to particles and molecular fluctuations. While the main focus of the software is to produce a point spread function (PSF) that can be used to model imaging, the results include propagation times so that the bandwidth of the optical channel when used for communication is also generated. Some commercial software packages exist that accomplish these results, but this report has presented a slightly different point of view, as well as a parallel processing MATLAB software package that implements the simulation.

APPENDIX
MATLAB CODE

```
%SCATTparallel
L. Estes
clc;clear;close all
%function[kout]=SCATT(b514,DR,NR,nrays,DTH,NPR)%SCATT.m
%seedoffset=input('seed offset [use 0 to (2^32-1)] =');
b514=0.037;
    bscale=input('bscale=');
    Nthreads=input('# of parallel threads =? ')
    nrays=input('total # of rays=');
    nrays=ceil(nrays/Nthreads');%for parallel processing
    DR=input('sphere radius step size =?');
    NR=input(' number of range steps=?');
    DTH=input('initial source cone angle (deg, 0+:180))= ?');
    b514=bscale*b514;
    %c530=b514+a;
        %rand('state',seed)
    DTH=DTH*pi/180;
    maxstep=4*ceil(b514*DR*NR+12*((b514*DR*NR)^0.5))
    Domega=2*pi*(1-cos(DTH));%cone steradians
    OTH=Domega/(2*pi);

tic
%matlabpool close force local
matlabpool close
%matlabpool open
matlabpool(Nthreads)
        parfor NPR = 1:Nthreads
            %c(:,i) = eig(rand(1000));
            %c(:,:,i)=stat2(1:i);
            %K(NPR)=SCATT(.04,17,3,10000,180,NPR)
            K(NPR)=SCATT(b514,DR,NR,nrays,DTH,NPR)
        end
 toc
       %x=1:10;
        %stat2(x)
         %function [mean,stdev] = stat2(x)
           %STAT Interesting statistics.
```

```
%backupparallel
%L. Estes
clc;clear
%K=backupfn(22,2,2)

%function[kout]=backupfn(DR,NR,NPR)

    Nthreads=input('# of parallel threads =? ')
    %nrays=input('total # of rays=');
    %nrays=ceil(nrays/Nthreads');%for prallel processing
    DR=input('sphere radius step size =?');
    NR=input(' number of range steps=?');


tic
%matlabpool close force local
matlabpool close
%matlabpool open
matlabpool(Nthreads)
        parfor NPR = 1:Nthreads
            %c(:,i) = eig(rand(1000));
            %c(:,:,i)=stat2(1:i);
            %K(NPR)=SCATT(.04,17,3,10000,180,NPR)
            K(NPR)=backupfn(DR,NR,NPR)
        end
 toc
```

```
%PSF789parallel.m
%L. Estes
clc
clear all
close all
format compact
a=input('absorption coef m^-1=? ')
Ns=input('number of range steps=? ')
DR=input('range step size m =?')
Nthreads=input('number of threads =')
r=(1:Ns)*DR;
ANGLEbarA=zeros(1,Ns);
ANGLErmsA=zeros(1,Ns);
toc1=zeros(1,Ns);
toc2=zeros(1,Ns);
toc3=zeros(1,Ns);
KdiffA=zeros(1,Ns);
KcoefA=zeros(1,Ns);
tic
for Nr=1:Ns

    Rr=Nr*DR
    tic
    for NPR=1:Nthreads
    Nfile=['RE789_',num2str(Rr),'m_',num2str(NPR),'threadHIT.mat'];
    load (Nfile)
    if NPR==1
        R789t=RE789;
    else
        R789t=[R789t;RE789];
    end
    end

    RE789=R789t;
    clear R789t
    nrays=nrays*Nthreads;
    toc

    %RE(ray pos vector, output ray unit vector, travel length in m', # of
    %scatterings, output angle relative to position vector), R=range'
    %N=range steps, seed, b514, nrays=launched ray number, TOC1=scatt
    %compute time, TOC2=backup time')
    c530=a+b514;
    sz=size(RE789);
    krays=sz(1);
    raypower=exp(-a*RE789(:,1));powsum=sum(raypower);
    ANGLEbarA(Nr)=sum(RE789(:,3).*raypower)/powsum;
    ANGLEBAR=ANGLEbarA(Nr);
    ANGLErmsA(Nr)=(sum((RE789(:,3).^2).*raypower)/powsum)^0.5;
    ANGLErms=ANGLErmsA(Nr);
    toc1(Nr)=TOC1;
    toc2(Nr)=TOC2;
    %get diffuse attenuation coef
    KdiffA(Nr)=(log(nrays/powsum))/R;%DiffuseAttenuationCoef
    DiffuseAttenuationCoef=KdiffA(Nr);
    KcoefA(Nr)=KdiffA(Nr)/a;
    Kcoef=KcoefA(Nr);
```

```
    %construct diffuse images assuming f/theta(deg)lens
    DTH=0.0547;%deg
    DTHairDeg=DTH*1.34;
    THETA=(0:DTH:90);
    LTH=length(THETA)
    DENOM=zeros(1,LTH-1);
    PDIF=0;
    for NL=1:LTH-1
        DENOM(NL)=2*pi*((THETA(NL+1)+THETA(NL))/2)*DTH;
    end
    Image=zeros(1,LTH-1);
    COHpower=0;
    Nnoscatterings=0;
    for K=1:krays
        if RE789(K,2)>0
            NTH=ceil(RE789(K,3)/DTH);
                if NTH==0
                    NTH=1
                end
            Image(NTH)=Image(NTH)+raypower(K)/DENOM(NTH);
            PDIF=PDIF+raypower(K);
            else
            COHpower=COHpower+raypower(K);
            Nnoscatterings=Nnoscatterings+1;
        end
    end
    COHpower=COHpower/nrays
    Nnoscatterings
    Image=Image/PDIF;
    ImageFinalOnSphereOneWattSource=Image*PDIF/nrays;
    savefile=['Image',num2str(Rr),'m_seed1_3.mat']
    TOC3=toc
    toc3(Nr)=TOC3;

save(savefile,'Image','DENOM','THETA','R','PDIF','nrays','ImageFinalOnSphereO
neWattSource','DiffuseAttenuationCoef','Kcoef','a','b514','ANGLErms','ANGLEBA
R','c530','COHpower','Nnoscatterings','DTHairDeg','TOC1','TOC2','TOC3')
    figure(Nr);plot(THETA(2:length(THETA)),Image);xlabel ('theta degrees')
    title(['PSF range =',num2str(R)]);grid on
    ylabel(['PSF range =',num2str(R)]);
%save Image500sd1 Image DENOM THETA R PDIF nrays
%ImageFinalOnSphereOneWattSource DiffuseAttenuationCoef Kcoef a b514 ANGLErms
ANGLEBAR AngleRMSfirstOrderScatt c530 COHpower Nnoscatterings DTHairDeg
    %figure(1600)
%plot(THETA(2:LTH),Image);xlabel('degrees');grid on
%ylabel('total normalized diffuse image radial slice')

end
toc
NF=0
if NF==0
figure(51);plot(r,KdiffA);grid on;xlabel('range');ylabel('Kdiff')
figure(52);plot(r,KcoefA);grid on;xlabel('range');ylabel('Kcoef')
figure(53);plot(r,ANGLEbarA);grid on;xlabel('range');ylabel('ANGLEbar deg')
figure(54);plot(r,ANGLErmsA);grid on;xlabel('range');ylabel('ANGLErms deg')
figure(55);plot(r,toc1);xlabel('range');ylabel('toc1 sec');grid on
figure(56);plot(r,toc2);xlabel('range');ylabel('toc2 sec');grid on
figure(57);plot(r,toc3);xlabel('range');ylabel('toc3 sec');grid on
save KdKcAbArSeed1 KdiffA KcoefA ANGLEbarA ANGLErmsA a b514 r toc1 toc2 toc3
end
```

```
function[kout]=SCATT(b514,DR,NR,nrays,DTH,NPR)%SCATT.m
%L. Estes
%format compact
%Pt spread function simulation of 3D scattering of a light  at
    %depth z propagating upward
    %loc=point location in [x y z]
    %dir=direction as unit vector [x y z]
%clc
%clear all
%close all

%Data from deep water at Autec by Petzold(24 deg 29.0 min N, 77 deg 33 min
west)
%c530=0.151; b514=0.037;%*10;
%a=0.114;%=c530-b514 all m-1
nw=1.34;%water index of refraction
na=1;%air index of refraction
ca=(3e8)/na;%m/sec speed of light in air
cw=(3e8)/nw;%m/sec speed of light in water
can=pi/180;nac=1/can;



load probclear% loads prob dist pb , angles th, VSF B, bscatt
lb=length(pb);
%Data from deep water at Autec by Petzold(24 deg 29.0 min N, 77 deg 33 min
%west)
%b514=0.037;%Petzold Autec clear
%c530=0.151;a=0.114;%=c530-b514 all m-1

% user input
    %seedoffset=input('seed offset [use 0 to (2^32-1)] =');
    %bscale=input('bscale=');
    %Nthreads=input('# of parallel threads =? ')
    %nrays=input('total # of rays=');
    %nrays=ceil(nrays/Nthreads');%for prallel processing
    %DR=input('sphere radius step size =?');
    %NR=input(' number of range steps=?')
    %DTH=input('initial source cone angle (deg, 0+:180))= ?');
    %b514=bscale*b514;
    %c530=b514+a;
        %rand('state',seed)
    DTH=DTH*pi/180;
    maxstep=2*ceil(b514*DR*NR+12*((b514*DR*NR)^0.5));
    Domega=2*pi*(1-cos(DTH));%cone steradians
    OTH=Domega/(2*pi);
    RE=zeros(nrays,9);  % ray results array
    %RE=[location vector,direction unit vector,length traveled, # of
    %scatterings, exit angle relative to position vector]
    xc=[1 0 0]; %unit x vector
    yc=[0 1 0]; %unit y vector
    zc=[0 0 1]; %unit z vector
   %matlabpool

RE=zeros(nrays,9);
seed=NPR;
```

```
rand('state',seed)

for N=1:NR
    tic;
     R=N*DR;
     %total path length to the surface
     k=0;   %hit the surface ray counter
     sz=size(RE);
    krays=sz(1);
   for m=1:krays
     re1=RE(m,1:3); % source ray location in meters
     re2=re1;
     r=norm(re2);
     if r>=R
         k=k+1;
         RE(k,:)=RE(m,:);
     else
         re1=re2;
         ST=RE(m,7);
         nend=0;

     for n=1:maxstep,

         if N==1 & n==1
             %get source direction
            phis=2*pi*rand;sp=sin(phis);cp=cos(phis);
            ths=acos(1-OTH*rand);st=sin(ths);ct=cos(ths);
            ne=[st*cp st*sp ct];
            RE(m,4:6)=ne;
            RE(m,8)=-1;%initialize # of scatterings
         else
             %get ne
             if n==1
                 ne=RE(m,4:6);
             end
             pang=rand;
             np = find(pb>=pang,1); % SBD replacement of while loop below
             %np=1;
             %while pb(np)<pang,
             %   np=np+1;
             %end
             if pang==1
                 theta=180;%degrees
             elseif np==1
                 theta=(th(1))*(pang)/(pb(1));
             else
                 theta=th(np-1)+(th(np)-th(np-1))*(pang-pb(np-1))/(pb(np)-
pb(np-1));
             end
```

```
                %theta
             theta=can*theta;
             % determine azimuth angle relative to current direction
             phi=2*pi*rand;
             %scatter
             %get unit vector in xc,ne plane perpendicular to ne
             np1dot = xc-ne*(dot(ne,xc));
             np1=np1dot/norm(np1dot);
             %get unit vector perp to np1,ne plane
             np2=cross(ne,np1);% unit vector perp to ne and np1
             % now scatter the ray
             ne=ne*cos(theta)+sin(theta)*(np1*cos(phi)+np2*sin(phi));
         end

         %get s
         s=(-log((1-rand)))/b514;%step size to scattering event
         ST=ST+s;
         %step
         re2=re1+s*ne; %step before scattering
         r=norm(re2,2);

           if r<R%test if surface has been reached
             re1=re2;
             %surface not reached
           else
             alpha=nac*real(acos(dot(ne,re2)/r));
             nend=1;
             k=k+1;
             nn=RE(m,8)+n;
             RE(k,:)=[re2,ne,ST,nn,alpha];

         end

          if nend==1,break,end

      end

     end
   end
   kout=k;
   TOC1=toc;
  %save range result
  RE=RE(1:k,:);
  savefile=['RE',num2str(R),'m_',num2str(NPR),'thread.mat'];
  save(savefile,'RE','R','N','seed','b514','nrays','TOC1')
end
```

```
function[kout]=backupfn(DR,NR,NPR)
%backupfn.m
%L. Estes
format compact
clc
%clear all
%close all
nac=180/pi;
%NR=input('number of ranges = ?')
%DR=input('range step size m =?')
for Nr=1:NR
    tic
    Rr=Nr*DR;
    Nfile=['RE',num2str(Rr),'m_',num2str(NPR),'thread.mat'];
    load(Nfile);
    %=RE(ray pos vector,output ray unit vector, travel length, #of
    %scatterings,output angle relative to position vector),R=range,N=range
    %step, seed,b514,nrayslaunced, TOC1=compute time
    sz=size(RE);
    krays=sz(1);
    RE789=zeros(krays,3);%=ST,n,alpha=distance,scatt #,angle deg

    for m=1:krays
     %back up to surface
             ST=RE(m,7);
             ne=RE(m,4:6);
             re=RE(m,1:3);
             r=norm(re);
             dnr=dot(ne,re);
             dels=dnr-(dnr^2-r^2+R^2)^0.5;
               % dels
             re=re-dels*ne;
             ST=ST-dels;%correct travel distance to surface
             %tim_of_flight=ST/cw;%time of flight to surface
             alpha=real((acos(dot(ne,re)/R))*nac);
             %account for absorption
             %raypower=raypower*(scatEff^(n-1))*exp(-ST*a); %account for
scatterer absorption scatEff
             %raypower=raypower*exp(-ST*a);
             %nend=1;
             RE(m,1:3)=re;
             RE(m,7)=ST;
             RE(m,9)=alpha;
    end
    TOC2=toc;
   % savefile=['RE',num2str(Rr),'m_seed1_3HIT.mat'];
    savefile=['RE',num2str(Rr),'m_',num2str(NPR),'threadHIT.mat'];
    save(savefile,'RE','R','N','seed','b514','nrays','TOC1','TOC2')
    RE789=RE(:,7:9);
    savefile2=['RE789_',num2str(Rr),'m_',num2str(NPR),'threadHIT.mat'];
    save(savefile2,'RE789','R','N','seed','b514','nrays','TOC1','TOC2')
    kout=krays;
end
```

The probclear .mat file is loaded from the SCATT function and loads the probability distribution pb, angles th, volume scattering function B, and scattering coefficient bscatt. The size and character of these variables are:

B	1x55	440	double
th	1x55	440	double
pb	1x55	440	double
bscatt	1x1	8	double

The variable B is the volume scattering function data obtained by Petzold* from deep water at AUTEC (24° 29.0' N, 77° 33' W). The variable values are

```
B = 53.1800   40.4200   30.7300   23.7400   18.1400   13.6000    9.9540
7.1790    5.1100    3.5910    2.4980    1.7190    1.1710    0.7758
0.5087    0.3340    0.2196    0.1446    0.0952    0.0628    0.0416
0.0204    0.0110    0.0062    0.0039    0.0027    0.0019    0.0014
0.0010    0.0008    0.0006    0.0005    0.0004    0.0003    0.0003
0.0003    0.0002    0.0002    0.0002    0.0002    0.0002    0.0002
0.0002    0.0003    0.0003    0.0003    0.0003    0.0003    0.0003
0.0003    0.0004    0.0004    0.0005    0.0005    0.0005

th= 0.1000    0.1260    0.1580    0.2000    0.2510    0.3160    0.3980
0.5010    0.6310    0.7940    1.0000    1.2590    1.5850    1.9950
2.5120    3.1620    3.9810    5.0120    6.3100    7.9430   10.0000
15.0000   20.0000   25.0000   30.0000   35.0000   40.0000   45.0000
50.0000   55.0000   60.0000   65.0000   70.0000   75.0000   80.0000
85.0000   90.0000   95.0000  100.0000  105.0000  110.0000  115.0000
120.0000  125.0000  130.0000  135.0000  140.0000  145.0000  150.0000
155.0000  160.0000  165.0000  170.0000  175.0000  180.0000

pb= 0.0139    0.0209    0.0292    0.0397    0.0521    0.0671    0.0847
0.1050    0.1281    0.1539    0.1826    0.2141    0.2482    0.2846
0.3226    0.3622    0.4033    0.4462    0.4910    0.5377    0.5865
0.6798    0.7473    0.7948    0.8288    0.8547    0.8753    0.8917
0.9048    0.9153    0.9239    0.9311    0.9372    0.9426    0.9473
0.9515    0.9553    0.9589    0.9623    0.9655    0.9687    0.9718
0.9748    0.9779    0.9809    0.9838    0.9865    0.9891    0.9915
0.9937    0.9957    0.9974    0.9988    0.9997    1.0000

bscatt= 0.0367  .
```

For the data presented the units are: B in 1/(steradian-meters), th in degrees, pb is unitless, and bscatt in 1/m.

---

*Curtis D. Mobley, *Light and Water Radiative Transfer in Natural Waters,* Academic Press, San Diego, 1994, pp. 110-111.

```
%petzoldclear.m
%L. Estes
%Creates probclear.mat
%Data from deep water at Autec by Petzold(24 deg 29.0min N, 77 deg 33min W)
%c(530)=0.151, b(514)=0.037, a=c(530)-b(514)=0.114 all m-1

close all
th=zeros(1,55);%angle in degrees
B=zeros(1,55);%m-1sr-1 volume scattering function
th(1:8)=[0.1,0.126,0.158,0.2,0.251,0.316,0.398,0.501];
B(1:8)=[5.318,4.042,3.073,2.374,1.814,1.360,0.9954,0.7179]*10;

th(9:16)=[0.631,0.794,1,1.259,1.585,1.995,2.512,3.162];
B(9:16)=[5.11,3.591,2.498,1.719,1.171,0.7758,0.5087,0.334];

th(17:24)=[3.981,5.012,6.31,7.943,10,15,20,25];
B(17:24)=[2.196,1.446,0.9522,0.6282,0.4162,0.2038,0.1099,0.06166]*0.1;

th(25:32)=[30,35,40,45,50,55,60,65];
B(25:32)=[3.888,2.68,1.899,1.372,1.02,0.7683,0.6028,0.4883]*0.001;

th(33:40)=[70,75,80,85,90,95,100,105];
B(33:40)=[4.069,3.457,3.019,2.681,2.459,2.315,2.239,2.225]*0.0001;

th(41:48)=[110,115,120,125,130,135,140,145];
B(41:48)=[2.239,2.265,2.339,2.505,2.629,2.662,2.749,2.896]*0.0001;

th(49:55)=[150,155,160,165,170,175,180];
B(49:55)=[3.088,3.304,3.627,4.073,4.671,4.845,5.019]*0.0001;

figure(1)
plot(th,log10(B));grid;xlabel('angle in degrees')
ylabel('log10 (Ptezold clear(AUTEC 24 deg 29.0min N, 77 deg 33min W)
scattering function m-1sr-1)')

b=zeros(1,55);%=integrated scattering coef
g=pi/180;
w=g*th;I=B.*sin(w);
b(1)=pi*I(1)*w(1);
for n=2:55,
    b(n)=b(n-1)+pi*(I(n)+I(n-1))*(w(n)-w(n-1));
end

figure(2)
plot(th,b);grid;xlabel('angle in degrees')
ylabel('integrated scat coef m-1 (Ptezold clear(AUTEC 24 deg 29.0min N, 77
deg 33min W)')
bscatt=b(55)
pb=b/bscatt;
save probclear pb th B bscatt
```

```
figure(3)
plot(th,pb);grid;xlabel('angle in degrees')
ylabel('probability distribution (Ptezold clear(AUTEC 24 deg 29.0min N, 77
deg 33min W) scattering function m-1sr-1)')

%calculate the rms theta
w=g*th;I2=B.*w.*w.*(sin(w))/bscatt;an2=zeros(1,55);

an2(1)=pi*I2(1)*w(1);
NMAX=55;
for n=2:NMAX,
    an2(n)=an2(n-1)+pi*(I2(n)+I2(n-1))*(w(n)-w(n-1));
end
anglerms=(180/pi)*((an2(NMAX))^0.5)%rms angle in degrees

%calculate the avg of cos(theta)
w=g*th;I3=B.*(cos(w)).*(sin(w))/bscatt;cs=zeros(1,55);

cs(1)=pi*I3(1)*w(1);
NMAX=55;
for n=2:NMAX,
    cs(n)=cs(n-1)+pi*(I3(n)+I3(n-1))*(w(n)-w(n-1));
end
costhetabar=cs(NMAX)%rms angle in degrees

%calculate the avg of (cos(theta))^2
w=g*th;I5=B.*((cos(w)).^2).*(sin(w))/bscatt;cs2=zeros(1,55);

cs2(1)=pi*I5(1)*w(1);
NMAX=55;
for n=2:NMAX,
    cs2(n)=cs2(n-1)+pi*(I5(n)+I5(n-1))*(w(n)-w(n-1));
end
costhetasqbar=cs2(NMAX)%rms angle in degrees


%calculate the avg theta
w=g*th;I4=B.*w.*(sin(w))/bscatt;an=zeros(1,55);

an(1)=pi*I4(1)*w(1);
NMAX=55;
for n=2:NMAX,
    an(n)=an(n-1)+pi*(I4(n)+I4(n-1))*(w(n)-w(n-1));
end
avgangle=(180/pi)*(an(NMAX))%avg angle in degrees
```

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Office of Naval Research (ONR-321MS—M. Wardlaw)	1
Defense Technical Information Center	1